ARITECH

Aritech

Addressable Fire

I/O Programming Guide

IPG2000

Revision 0.2, March 2002

an interlogix company.

Aritech is an Interlogix company.

COPYRIGHT

© 2001 Interlogix B.V.. All rights reserved. Interlogix B.V. grants the right to reprint this document for internal use only. Interlogix B.V. reserves the right to change information without notice.

CONTENTS

1. Introduction 4

1.1. Scope 4

2. How to use this guide 4

2.1. Inputs 5
2.2. Outputs 5
2.3. Logic 5

3. Logic Programming 6

3.1. Basic Logic 6
3.1.1. = ... 6
3.2. Boolean Logic 7
3.2.1. And 7
3.2.2. Or 7
3.3. Rules of the Aritech logic programming 7

4. Switching Diagrams 10

5. Input Definition 13

5.1. Input Types 14
5.1.1. GENERAL 14
5.1.2. ZONE 15
5.1.3. AREA 15
5.1.4. ADJ. AREA 15
5.1.5. INTERNAL 16
5.1.6. TIME 16
5.1.7. DEVICE INPUT 16
5.1.8. DEVICE 16
5.1.9. NETWORK 16
5.1.10. ACTION 17
5.1.11. CL DEVICE 17
5.1.12. DATE 17

6. Output Definition 18

6.1. Outputs 18
6.2. Output Types 19
6.2.1. GENERAL 19
6.2.2. ZONE 19
6.2.3. AREA 19
6.2.4. INTERNAL 20
6.2.5. DEVICE OUTPUT 20
6.2.6. INTERNAL SUPERVISED 21
6.2.7. DEVICE SUPERVISED 21
6.2.8. NETWORK 22
6.2.9. CURRENT LOOP DEVICE 22
6.2.10. CURRENT LOOP SUPERVISED 22
6.2.11. EVENT 22
6.2.12. ACTION 22

7. Basic Programming 24

8. Advanced Programming 32

1. INTRODUCTION

1.1. Scope

This manual explains how to set-up and test the Input Table, Output Table and Logic Table programming for the Aritech Addressable and Analogue Addressable fire detection panels.

Other reference manuals that may be consulted are:

FP2000/1200/1100 Reference Guide

FP2000/1200/1100 Installation and Commissioning Manual

FP2000/1200/1100 User's Manual

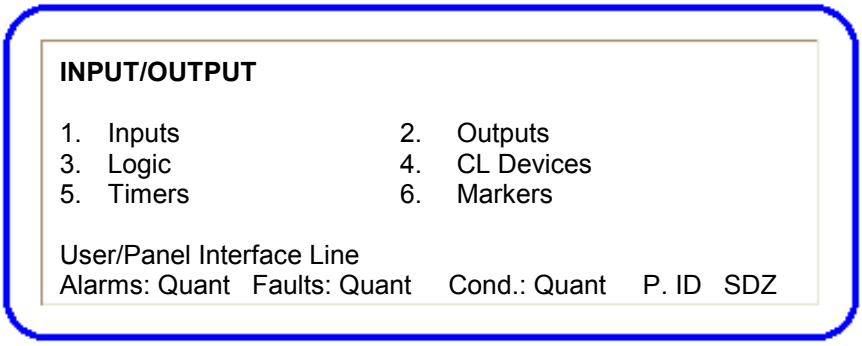
900 Series Installation Guide

2000 Series Installation Guide

2. HOW TO USE THIS GUIDE

Logic programming on the Aritech fire panel is divided into 3 main categories: Inputs, Outputs and Logic. Inputs can be defined as the ***'cause'*** of an action, outputs as the ***'effect'*** of this action, whilst the logic is the ***mechanism*** to connect the cause to the effect.

Figure 1: Input/Output Menu

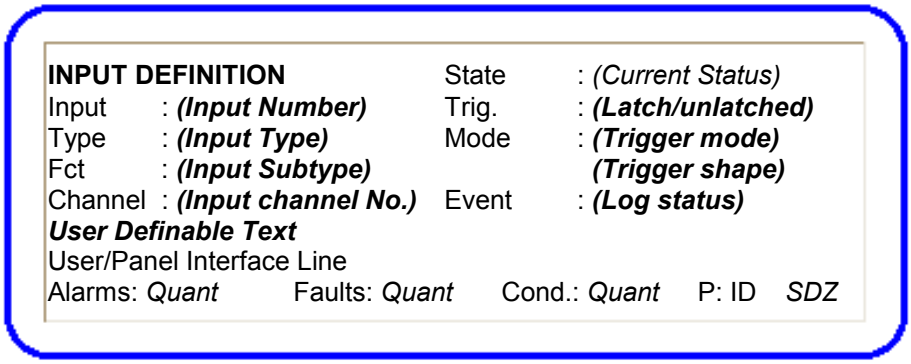


The functionality of these tables is indicated on the fire panel LCD display as follows:

— **Note:** The boldface items indicate the important information that the user has to supply for the panel. Depending on the specific input, output or logic type, some fields may not be required or may not be available.

2.1. Inputs

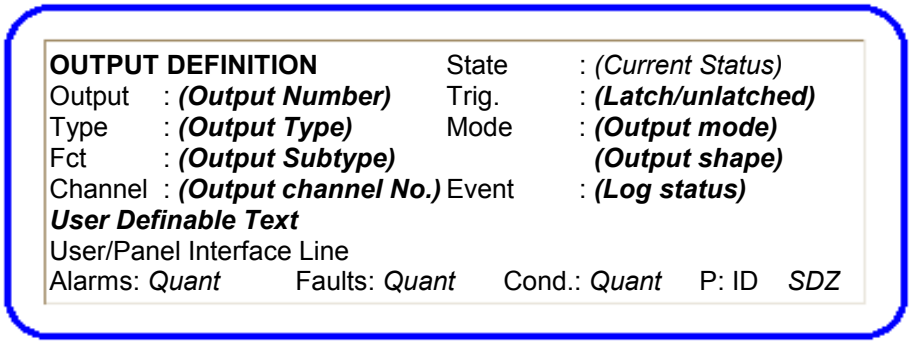
Figure 2: Input Menu



Refer to section 4 Switching Diagrams

2.2. Outputs

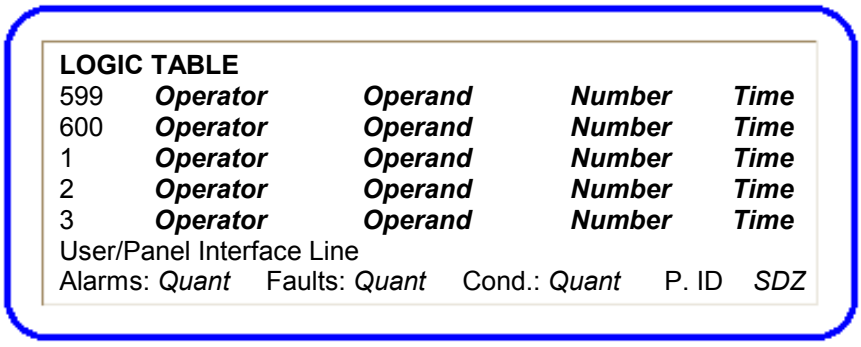
Figure 3: Output Menu



Refer to section 6 Output Definition

2.3. Logic

Figure 4: Logic Table



3. LOGIC PROGRAMMING

The logic programming of the Aritech fire panels follow a simple cause and effect sequence i.e. when something happens (**cause**), we also want *something else* (**effect**) to happen. For the panel, the cause will be defined as an ***Input***, whilst the effect is defined as an ***Output***.

The inputs (causes) for the panel are defined in a table called the ***Input Table***. The size of this table is adjustable in the memory configuration of the panel. By default there will be 150 inputs available to the user, but this may be expanded as required, to a maximum number of 999. The outputs (effects) for the panel are defined in a table called the ***Output Table***. The size of this table is the same as for the Input Table, and is therefore also adjustable in the memory configuration of the panel. By default there will be 150 available outputs for the user, but this may be expanded as required, to a maximum number of 999.

In order to link the specific cause in the Input Table to a specific effect in the Output Table, some sort of mechanism needs to be used. The **mechanism** for linking cause to effect (Inputs to Outputs) is called a ***Logic Table***. The logic table has the function of taking the specific input number defined in the input table, and linking it to a specific output number in the table. The size of this table is also adjustable in the memory configuration of the panel. By default there will be 600 available lines of logic, but this may be expanded as required, to a maximum number of 1999.

In addition to just linking inputs to outputs, certain special functions may also be implemented in the logic to allow for complex functionality and timing functions. These are done in the form of **Boolean logic, markers and timers**. These functions will all be discussed in greater detail later on in this document.

- **Note:** Remember that the numbers in the sections below refer to the table number in the input and output table. At this point we are not interested in what is actually entered in these tables, since for the operation of the logic it really makes no difference. We will discuss the Inputs and Outputs in the table later on in this manual.

- **Note:** The ' **true**' and '**false**' flags of the inputs and outputs are indicated in the top right-hand corner of every input and output screen on the panel. (Refer to 'state' in Figure 2 and Figure 3 above.) This does not necessarily mean that the input is on or of, but rather whether it is triggered or not. This will become clearer when we discuss the inputs and outputs later on in this manual.

3.1. Basic Logic

Basic Logic programming consist of the simple '=' function. Variations of this function are available, but they will be treated in the 'Advanced Programming' section of this manual. (See Section 7)

3.1.1. =

The '=' function is a function that allows for an output (effect) to be dependent on an input (cause), and will only act on the state of this input.

E.g. (Input 1)= Output 1

Here the following scenarios may take place:

Input 1 is **false**	then Output 1 will be **false**
Input 1 is **true**	then Output 1 will be **true**

3.2. Boolean Logic

Apart from the simple '=' function, the fire panel employs just two basic logic functions, 'and' and 'or'. Variations of all these functions are available, but they will be treated in the 'Advanced Programming' section of this manual. (See Section 7)

3.2.1. And

The 'and' function is a function that allows for an output to be independent on the individual inputs, but will act on the ***combined*** state of all the inputs linked to this output.

E.g. (Input 1 and Input 2)= Output 1

Here the following scenarios may take place:

Input 1 is false	Input 2 is false	then Output 1 will be false
Input 1 is false	Input 2 is true	then Output 1 will be false
Input 1 is true	Input 2 is false	then Output 1 will be false
Input 1 is true	Input 2 is true	then Output 1 will be true

3.2.2. Or

The 'or' function is a function that allows for an output to be dependent on every individual input, and will act on the individual state of every inputs linked to this output.

E.g. (Input 1 or Input 2)= Output 1

Here the following scenarios may take place:

Input 1 is false	Input 2 is false	then Output 1 will be false
Input 1 is false	Input 2 is true	then Output 1 will be true
Input 1 is true	Input 2 is false	then Output 1 will be true
Input 1 is true	Input 2 is true	then Output 1 will be true

3.3. Rules of the Aritech logic programming

There are only a few rules that apply to the fire panel's input, output and logic table. Some of the rules are essential for the correct operation of the panel, but others are recommended only for ease of use and debugging. There are, as always, also some exceptions to some rules, but these will be handled separately in the Advance Programming section (Section 7) of this manual.

The basic programming rules are as follows:

1. A specific input should only be defined once in the Input table. For example:

   If I want to use a specific input to trigger more than one output, I should use the logic table to link the input to these outputs in stead of defining the input more than once in the input table to match the number of outputs.

   This is not necessarily flagged as a fault condition UNLESS the inputs are configured differently in the options for the 2 inputs. It is however recommended avoiding this configuration to eliminate later problems.

2. A specific output should only be defined once in the Output Table. For example:

   If I want to use more than one input to trigger a single output, I should use the logic table to link these inputs to the output in stead of defining the output more than once in the output table.

   This is not necessarily flagged as a fault condition UNLESS the outputs are configured differently in the options for the 2 outputs. It is however recommended avoiding this configuration to eliminate later problems.

3. Every logic function should start with a '**(**'. For example:

```
(       Input   1
)=      Output  1
```

is considered to be a complete function starting with a '('.

If not followed, this will be flagged as a fault condition and will cause the panel to suspend all logic functions until the error is corrected. (Refer to the Faultfinding Guide for further information on this fault.)

4. A logic function always ends after a '=' sign. For example:

```
(       Input   1
)=      Output  1
```

This is considered to be a complete function ending with a '='. Any function after this has to again comply with rule 3 above. If not, this will be flagged as a fault condition and will cause the panel to suspend all logic functions until the error is corrected. (Refer to the Faultfinding Guide for further information on this fault.)

5. In a single logic function, there should be just as many '**)**' (closed-brackets) as there are '**(**' (open-brackets). For example:

Complex functions (functions within functions) are allowed in the logic (See section 7) such as

```
(       Input   1
or(     Input   2
and     Input   3
)
)=      Output  1
```

This function consists of two '(' (open-brackets) and two ')' (closed-brackets) and is therefore complete. If a bracket is left out, this will be flagged as a fault condition and will cause the panel to suspend all logic functions until the error is corrected.

6. An output should never be defined as an output function more than once in the logic table. For example:

If a situation situations arises where more than one inputs have to trigger a single output, the two inputs should be linked to the output in the logic using a single function with 'and' or 'or'. If every input is linked to this output separately, only the last function (the function with the highest logic number referring to this output) will activate the output.

This is not flagged as a fault condition and can cause rather big programming problems.

7. Outputs should always be inserted in the logic table as outputs in numerical sequence. For example:

```
(       Input   1
)=      Output  1
(       Input   2
)=      Output  5
…
(       Input   7
)=      Output  10
```

This functions consist of more than three functions, but the ')= Output x' operand is always in numerical sequence i.e. the highest number is always below its predecessor. This is not flagged as a fault condition and will not cause any problems, but it will assist with the implementation of rule 6 above, as well as help greatly in the debugging of the logic should this ever be required.

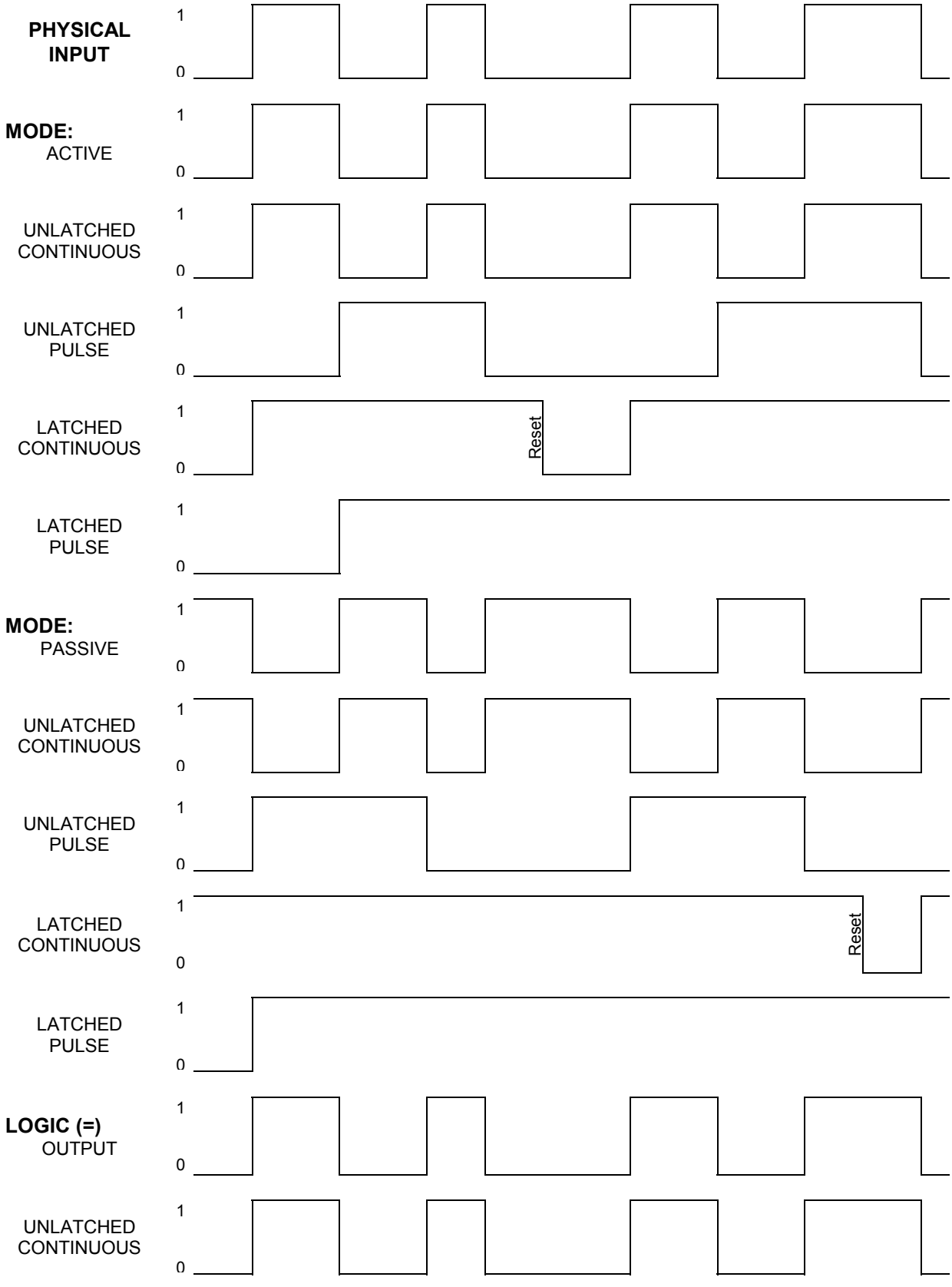
8. The final line of logic in the logic table should have an '**end**' statement in the operator field. For example:

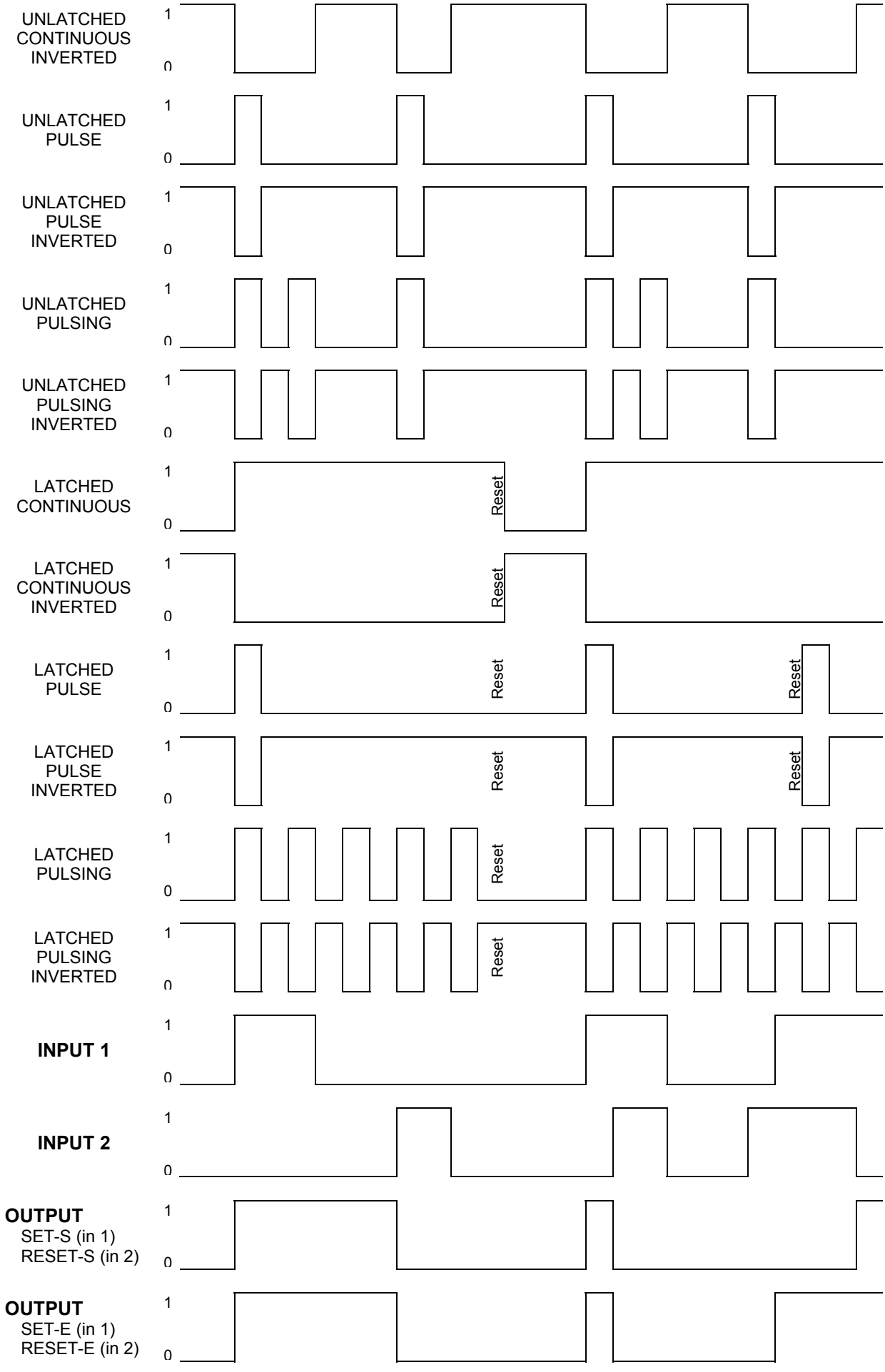
```
(       Input   1
)=      Output  1
(       Input   2
)=      Output  5
end
```

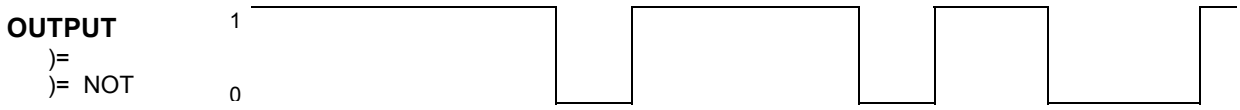
This is *not flagged* as a fault condition and will not cause any problems. It just helps the panel not to scan all blank lines in the logic table to find no further executable logic, and therefore speeds up the processing in the panel.

4. SWITCHING DIAGRAMS

The following table provides an overview of the shape of an input and an output in the panel when used with the various configuration options in the logic as described in sections 5, 6 and 7 in this document.



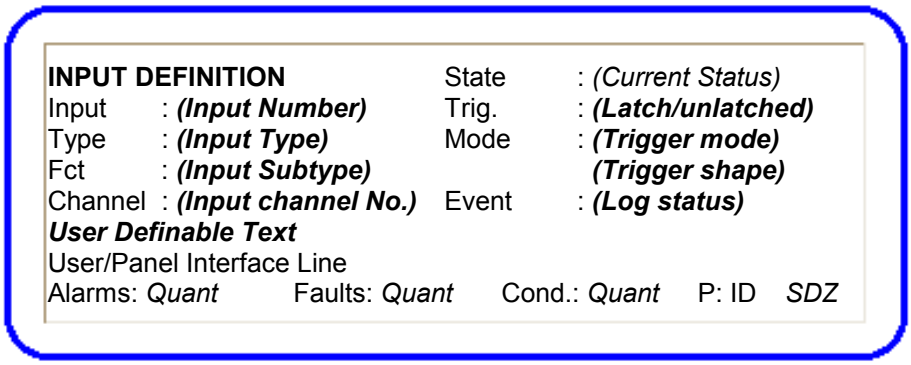

UNLATCHED CONTINUOUS INVERTED
UNLATCHED PULSE
UNLATCHED PULSE INVERTED
UNLATCHED PULSING
UNLATCHED PULSING INVERTED
LATCHED CONTINUOUS
LATCHED CONTINUOUS INVERTED
LATCHED PULSE
LATCHED PULSE INVERTED
LATCHED PULSING
LATCHED PULSING INVERTED
INPUT 1
INPUT 2
OUTPUT
SET-S (in 1)
RESET-S (in 2)
OUTPUT
SET-E (in 1)
RESET-E (in 2)
1
0
Reset


OUTPUT
)=
)= NOT
1
0

5. INPUT DEFINITION

Defining an input will define the ***cause*** of the function.

Figure 5: Input Menu



Input Number:	This number only defines the position of the input in the input table. The size of the input table can be adjusted in the memory configuration of the panel. The default size will allow 150 input entries, but a maximum of 999 inputs locations may be configured. (Refer to the FP2000 Reference Guide for further information.)
Input Type:	This defines the input group that this input belongs to. Each input group will have different subtypes. The input group is only used to 'sort' the input types to make location of the required input easier.
Input Sub-Type:	This defines the specific function of the selected input group.
Input Channel Number:	(Optional – depending on the input type) Should the specific input unit allow for more than one physical input, the specific input used will have to be defined here.
User Definable Text:	(Optional – depending on the input type) Text that allows the user to describe the function of this input. Some inputs have predefined or self-explanatory functions, and will therefore have no definable user text.
Current Status:	Indicates whether this input is currently '**TRUE**' or '**FALSE**' – this may be used for debugging and diagnostic purposes. This state is updated every time the panel reads the input, at least once ever 10 seconds.
Latched/Unlatched:	A selection to allow the programmer to decide whether this input will follow the status of the monitoring point (unlatched), or whether it will 'latch' – remain in the 'TRUE' state once activated.
Trigger Mode:	The specific function of this input that will cause the input to change to the 'TRUE' state e.g. activated, passive, open circuit or short circuit.
Trigger Shape:	The shape of the input to be expected e.g. a transition from FALSE to TRUE and back to FALSE (pulse mode), or simply a transition from FALSE to TRUE (continuous mode).
Log Status:	(Optional – depending on the input type) The programmer may decide how the event processor of the panel handles this event. Will the fact that the input changes to the TRUE state be logged in the event log or not, and if so, will it have any effect on the operation of the panel i.e. will it cause an alarm. Some inputs have pre-defined functions, and therefore are not user definable.

5.1. Input Types

The various input types are simply groups of inputs to make finding of the required input easier. All inputs have been **grouped by location** in the panel, as can be seen from the functions below.

— **Note:** Remember that some of these inputs have defined functionality. They will automatically do what they are supposed to be doing. What we will do with our input/output and logic programming functions will be **in addition to** what the panel will normally do when these inputs activate.

5.1.1. GENERAL

These inputs form part of the 'general' operating functions of the panel.

☞ *Explanations for the definitions of these inputs and when they will appear on the panel may be found in the Aritech Analogue Addressable Faultfinding Guide.*

The subtypes are: (indicating what particular general status is of interest)

Access Fault
Battery Disconnected
Battery Test Failed
Charger Fault
Current Loop Device Fault
Coincidence
Common Condition
Common Fault
Common Fire
Disable
Earth Fault
Emulation Disconnected
External Fault
External Fire
External Supply Fault
Fault-routing Disabled
Fault-routing Fault
Fault-routing Test
Fire Brigade Disabled
Fire Brigade Fault
Fire Brigade Test
Fire-protection Disabled
Fire-protection Fault
Fire-protection Test
Global Repeater Fault
Hardware Fault
Local Repeater Fault
Low Battery
Mains Disconnected
Maintenance Fault
Memory unlocked
Modem Fault
Panel Fault
Printer Disconnected

Service Switch On
Soak Test
Sounder Disabled
Sounder Fault
Sounder Test
Supply Fault
System Fault
Tamper Switch
Test
VDU Disconnected
Zone Test

5.1.2. ZONE

These inputs form part of the detection zone functions of the panel. A zone is defined as ***a group of detection devices***.

The subtypes are: (indicating what particular zone status is of interest)

Fire
Fault
Coincidence
Condition
Disable

5.1.3. AREA

These inputs form part of the detection area functions of the panel. An area is defined as ***a group of detection zones***.

The subtypes are: (indicating what particular area status is of interest)

Fire
Fault
Coincidence
Condition
Disable

5.1.4. ADJ. AREA

These inputs form part of the detection area functions of the panel. An adjacent area is defined as ***an area next to the area that I am interested in***.

The subtypes are: (indicating what particular status of any adjacent area to my area is of interest)

Fire
Fault
Coincidence
Condition
Disable

5.1.5. INTERNAL

These inputs form part of the 'internal' inputs of the panel. These inputs are located on the FEP2000 and the SD2000 PCB's on the back panel electronics of the panel. (Refer to the various Installation and Commissioning manuals for more information on these inputs.)

The subtypes are: (indicating what PCB this input is located on, and what particular input on this PCB is of interest – both the FEP2000 and the SD2000 have multiple inputs.)

Board Number:
Input Channel:

5.1.6. TIME

These inputs are generated from the internal, real-time clock of the panel.

The subtypes are: (in hour, minute format)

Time: hh mm
Day: day of the week

5.1.7. DEVICE INPUT

These inputs are generated by Input Units connected to the detection loop of the panel.

— **Note:** Do not confuse these with the inputs generated by the detection devices connected to the detection loop of the panel! (See section 5.1.8)

The subtypes are: (indicating where the I/O Unit is and what input channel - in cases where a multi-channel I/O unit is used – is of interest.)

Device Number:
Input Channel:

5.1.8. DEVICE

These inputs are generated by the detection devices connected to the detection loop of the panel.

— **Note:** Do not confuse these with the inputs generated by the Input Units connected to the detection loop of the panel! (See section 5.1.7)

The subtypes are: (indicating where the device is and what state of this device is of interest.)

Device Number:
Cause: (Fire, Fault or Condition)

5.1.9. NETWORK

These inputs are located on the Fire Panel Network.

— **Note:** By definition this means that there will also be an output somewhere in the system onto the Fire Panel Network. Using this function will allow inputs and outputs to be sent between fire panel nodes on the same network.

The subtypes are: (Indicating where the output is generated in the network.)

Node ID: xx/xx
Output: 1…999

5.1.10. ACTION

These inputs are generated by an action, either performed by the panel itself, or by a system operator.

The subtypes are: (Indicating what action it is that is of interest.)

Day Mode
Zones On
School Bells on
Silence Buzzer
Restart
Reset
Access enabled
Event buffer full
Maintenance reminder
Key switch enabled
Event Buffer cleared
Fire Brigade signal
Fire Brigade stop
Fire Brigade delay on
Sounder on
Sounder silenced
Sounder delay on
Fprot delay on
Fprot on
Fltrt on
Fltrt delay on

5.1.11. CL DEVICE

These inputs are located on the devices connected to the Current Loop network of the panel.

The subtypes are: (Indicating what current loop device the input is coming from and what input on this device is of interest – all current loop devices have multiple inputs.)

CL Device Number:
Input Channel:

5.1.12. DATE

These inputs are generated from the internal, real-time clock of the panel. This input will normally become active at 00:00 on the date that was programmed here.

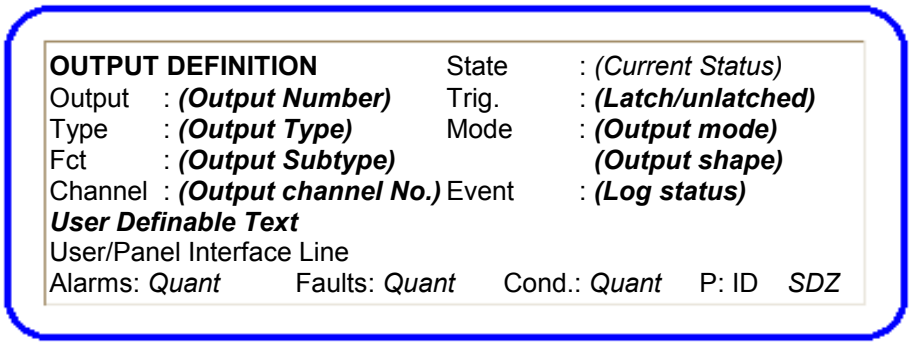
The subtypes are: (in day month year format)

Date: dd mm yy

6. OUTPUT DEFINITION

6.1. Outputs

Figure 6: Output Menu



Defining an output will define the ***effect*** of the function.

Output Number:	This number defines the position of the output in the output table. The size of the output table can be adjusted in the memory configuration of the panel. The default size will allow 150 Output entries, but a maximum of 999 output locations may be configured. (Refer to the FP2000 Reference Guide for further information)
Output Type:	This defines the output group that this output belongs to. Each output group will have different subtypes.
Output Sub-Type:	This defines the specific function of the selected output group.
Output Channel Number:	(Optional – depending on the output type) Should the specific output unit allow for more than one physical output, the specific output used will have to be defined here.
User Definable Text:	(Optional – depending on the output type) Text that allows the user to describe the function of this output. Some outputs have predefines or self-explanatory functions, and will therefore have no definable user text.
Current Status:	Indicates whether this output is currently 'TRUE' or 'FALSE' – this may be used for debugging and diagnostic purposes. This state is updated every time the panel reads the output.
Latched/Unlatched:	A selection to allow the programmer to decide whether this output will follow the status of the monitoring point (unlatched), or whether it will 'latch' – remain in the 'TRUE' state once activated.
Trigger Mode:	The specific function of this output that will cause the output to change to the 'TRUE' state.
Trigger Shape:	The shape of the output to be expected e.g. a transition from FALSE to TRUE and back to FALSE (pulse), or simply a transition from FALSE to TRUE (continuous).
Log Status:	(Optional – depending on the output type) The programmer may decide how the event processor of the panel handles this event. Will the fact that the output changes to the TRUE state be logged in the event log or not, and if so, will it have any effect on the operation of the panel i.e. will it cause an alarm. Some outputs have pre-defined functions, and therefore are not user definable.

6.2. Output Types

The various output types are simply groups of outputs to make finding of the required output easier. All outputs have been **grouped by location** in the panel, as can be seen from the functions below.

6.2.1. GENERAL

These outputs form part of the ‘general’ operating functions of the panel.

The subtypes are: (indicating what particular general status effect will be caused)

Common Condition
Common Fault
Common Fire
External Fault
External Fire
External Supply Fault
Fault-routing Disabled
Fault-routing Test
Fire Brigade Disabled
Fire Brigade Test
Fire-protection Disabled
Fire-protection Test
Service Switch On
Sounder Disabled
Sounder Test
Hardware Fault
Tamper Switch

6.2.2. ZONE

These outputs form part of the ‘zone’ operating functions of the panel.

The subtypes are: (indicating what particular zone status effect will be caused)

Fire MCP
Fire Auto
Fault
Coincidence
Condition
Disable
Pre-Alarm

6.2.3. AREA

These outputs form part of the ‘area’ operating functions of the panel.

The subtypes are: (indicating what particular area status effect will be caused)

Fire
Fault
Coincidence
Condition
Disable
Pre-Alarm

6.2.4. INTERNAL

These outputs form part of the 'internal' outputs of the panel. These outputs are located on the RB2016 and the SD2000 PCB's on the back panel electronics of the panel. (Refer to the various Installation and Commissioning manuals for more information on these outputs.)

— **Note:** Do not confuse these with the supervised internal outputs located on the SD2000 and SD1200 PCB's! (See section 6.2.6)

The subtypes are: (indicating what PCB this output is located on, and what particular output on this PCB is of interest – both the RB2016 and the SD2000 have multiple outputs.)

Board number:
Output Channel:
Link to:
Sounders - (General, Zone or Area)
Fire Brigade - (General, Zone or Area)
Fault Routing - (General, Zone or Area)
Fire Protection - (General, Zone or Area)
Logic

6.2.5. DEVICE OUTPUT

These outputs are located on the ***Output Units*** connected to the detection loop of the panel.

— **Note:** Do not confuse these with the supervised outputs located on some I/O units! (See section 6.2.7)

The subtypes are: (indicating where the I/O Unit is and what output channel - in cases where a multi-channel I/O unit is used – is of interest.)

Device Number:
Output Channel:
Link to:
Sounders - (General, Zone or Area)
Fire Brigade - (General, Zone or Area)
Fault Routing - (General, Zone or Area)
Fire Protection - (General, Zone or Area)
Logic

6.2.6. INTERNAL SUPERVISED

These outputs form part of the 'internal' outputs of the panel. These outputs are located on the SD2000 or SD1200 PCB's on the back panel electronics of the panel. (Refer to the various Installation and Commissioning manuals for more information on these outputs.)

— **Note:** Do not confuse these with the un-supervised internal outputs located on the RB2016 and SD2000 PCB's! (See section 6.2.4)

— Even though these relays are allowed to be used for user programming, always remember that they have pre-programmed functions. These functions are determined by regulations and are not removable. Any user function will simply be 'or'ed with the pre-programmed functions, leading to undesirable effects in most cases. It is therefore highly recommended that these relays NOT BE USED FOR USER PROGRAMMING FUNCTIONS.

The subtypes are: (indicating what PCB this output is located on, and what particular output on this PCB is of interest – both the SD2000 and the SD1200 have multiple outputs.)

Board Number:
Output Channel:
Link to:
- Sounders - (General, Zone or Area)
- Fire Brigade - (General, Zone or Area)
- Fault Routing - (General, Zone or Area)
- Fire Protection - (General, Zone or Area)
- Logic

6.2.7. DEVICE SUPERVISED

These outputs are located on the ***Output Units*** connected to the detection loop of the panel.

— **Note:** Do not confuse these with the un-supervised outputs located on some I/O units! (See section 6.2.5)

The subtypes are: (indicating where the I/O Unit is and what output channel is of interest. All supervised I/O only have one channel i.e. channel 1)

Device Number:
Output Channel:
Link to:
- Sounders - (General, Zone or Area)
- Fire Brigade - (General, Zone or Area)
- Fault Routing - (General, Zone or Area)
- Fire Protection - (General, Zone or Area)
- Logic

6.2.8. NETWORK

These outputs are located on the Fire Panel Network.

— By definition this means that there will also be an input somewhere in the system from the Fire Panel Network. Using this function will allow inputs and outputs to be sent between fire panel nodes on the same network.

The subtypes are: (Indicating where the input is located in the network.)

Node ID: xx/xx
Input Number: 1…999

6.2.9. CURRENT LOOP DEVICE

These outputs are located on the devices connected to the Current Loop network of the panel.

The subtypes are: (Indicating what current loop device the output is located on, and what output on this device is of interest – all current loop devices have multiple outputs.)

Current Loop Device Number:
Output Number:
Link to:
Sounders - (General, Zone or Area)
Fire Brigade - (General, Zone or Area)
Fault Routing - (General, Zone or Area)
Fire Protection - (General, Zone or Area)
Logic

6.2.10. CURRENT LOOP SUPERVISED

These outputs are located on the devices connected to the Current Loop network of the panel.

The subtypes are: (Indicating what current loop device the output is located on, and what output on this device is of interest – all current loop devices have multiple outputs.)

— **Note:** Aritech have not produced any supervised current loop devices. This output definition is therefore NOT CURRENTLY SUPPORTED. It is only for future use.

Current Loop Device Number:
Output Number:
Link to:
Sounders - (General, Zone or Area)
Fire Brigade - (General, Zone or Area)
Fault Routing - (General, Zone or Area)
Fire Protection - (General, Zone or Area)
Logic

6.2.11. EVENT

These outputs allow the user to create text messages of 40 characters in length. These messages may be displayed on the LCD display of the panel, or simply be logged in the event log of the panel.

6.2.12. ACTION

These outputs will cause an action to be performed on the panel.

The subtypes are: (Indicating what action will be performed.)

Call on Line 1
Call on Line 2
Call on Line 3
Call on Line 4
Day Mode
Fault-routing Delay On
Fault-routing Off
Fault-routing On
Fire Brigade Delay On
Fire Brigade Signalled
Fire Brigade Stopped
Fire-protection Delay On
Fire-protection Off
Fire-protection On
Key Switch Unlocked
Reset
Restart
Schoolbells On
Silence Buzzer
Sounder Delay On
Sounder On
Sounder Silenced
Synchronise Time
Zones on

7. BASIC PROGRAMMING

All logic programming need not be complex, and in fact in most instances the programming in the field is rather straightforward. The following two sections will explain some of the usage of this facility in the panel with the help of some practical examples.

Example 1: Activating Outputs 'linked to' the Sounders or Fire Brigade outputs of the panel

Scenario: A basic fire installation with, in addition to the normal smoke detection devices, sounders or sounder controllers connected to the detection loops of the panel. The sounders must activate whenever a fire exists anywhere in the system.

Normally the panel will, depending of course on the operating mode (EN, EP, VdS or NEN) activate the common sounder output inside the panel automatically when a fire alarm exists on the panel. It will however not activate any outputs on the detection loops unless programmed to do so.

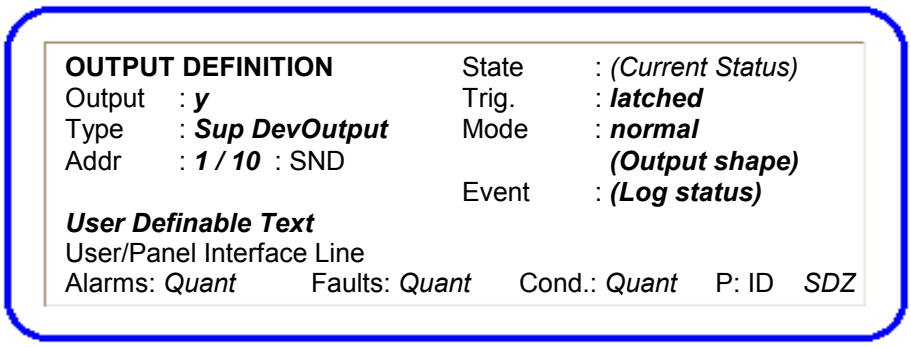
Equipment: Lets say we have two loop-powered sounders connected at address loop 1 device 10, and loop 2 device 100.

Programming: We will use the special function of the panel to simplify this programming. We already know that we have a sounder output that will activate whenever we have a fire. Why can we not link our loop-powered sounders to this output? We can!

Inputs: Not required

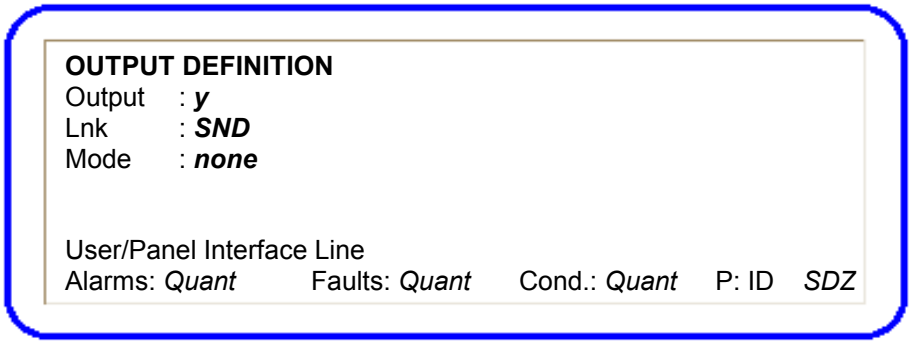
Outputs: (Refer to section 6.2.7 DEVICE SUPERVISED.) Note that the sounder has to be enabled in the device menus first.

Figure 7: Supervised Device Output Menu



and on the next screen…

Figure 8: Supervised Device Output Menu (2)



So, what we have done is we have successfully linked the sounders on Loop 1 Device 10 to the common sounder control of the panel. When there is a condition on the panel that would normally activate the common sounder output inside the panel, our loop-powered sounder will be activated as well. The sounder control on the front of the panel will also service this programmed sounder, so we will also be able to silence the sounder in the filed using these controls.

Set up another output similar to this one for the sounder at Loop 2 Device 100. Now the panel controls both sounders.

In Figure 8 above one can see that we did not use any 'mode' configuration. The mode configuration may be used in instances where I do not want my sounders to activate when there is a fire anywhere in the system, but only when there is a fire in the zone or area serviced by this sounder. I con then use this 'mode ' functionality for linking a sounder to a specific zone or area. The sounder is still controlled from the sounder output of the panel, but now only activates when a fire exists in that zone or area set in the 'mode' programming.

Any outputs in the system may be linked in this way, and controlled by the fire brigade and sounder controls on the fact of the panel.

Example 2: Activating an output on an input/output unit

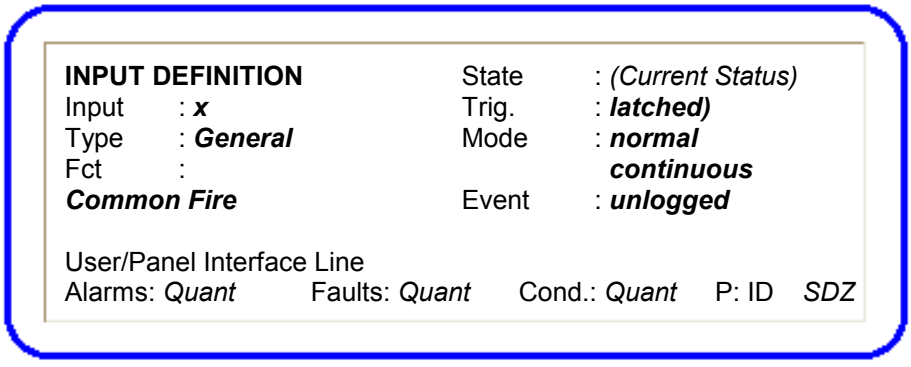
Scenario: A basic fire installation with, in addition to the normal smoke detection devices, output units connected to the detection loops of the panel. This output is used to hold open a fire door, and must therefore de-activate whenever a fire exists anywhere in the system.

Equipment: Lets say we have an Output Unit connected at address Loop 1 device 10.

Programming: We need to know when we want the output to de-activate, i.e. what event will we use (cause) that will cause this output to switch (effect). Once we have this, we can continue…

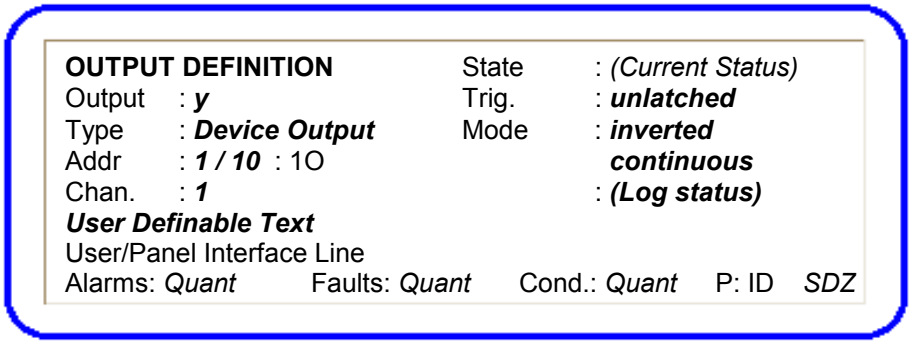
Inputs: Any fire in the system. (Refer to section 5.1.1 GENERAL)

Figure 9: General Common Fire Input



Outputs: (Refer to section 6.2.5 DEVICE OUTPUT.) Note that the Output device has to be enabled in the device menu first. Don't forget that this output must be active all the time, and deactivate when the event occurs. We can accomplish this very easily by simply inverting the 'mode' for this output in the output programming menus.

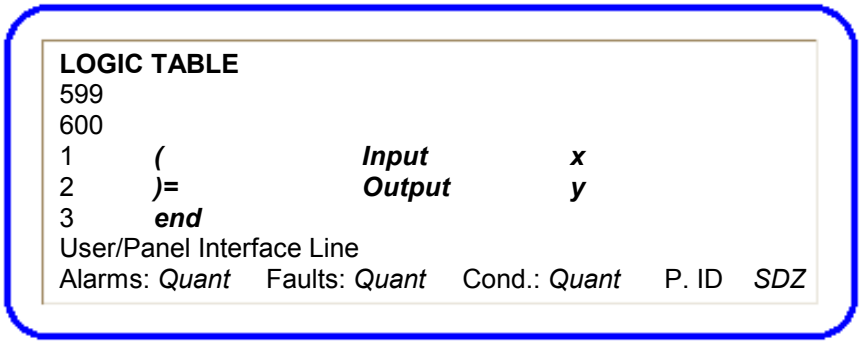
Figure 10: Device Output Menu



NOTE: If we want, we may even add user text to this output to describe what its function is. This way it will always know what this relay is in fact used for.

Logic: Now we have to link our 'cause' to our 'effect'.

Figure 11: Input x = Output y



What does this say in English?

If the input at input table location x is true
then make the output at output location y true

So, in our case: If there is a common fire condition
then switch the output on device loop 1 device 10

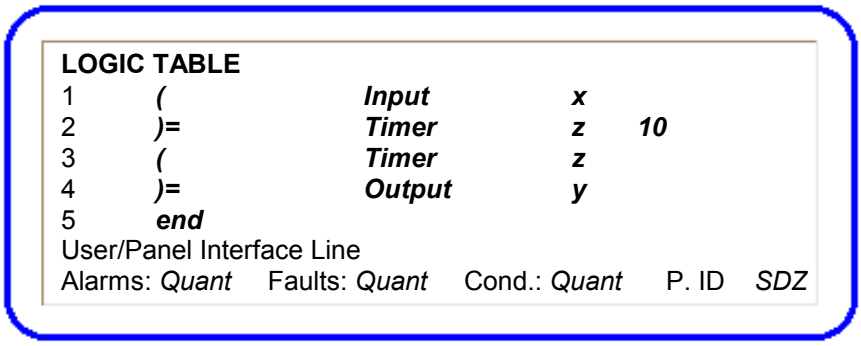
Simple logic as we see above, has the effect of activating our output immediately when there is a fire. Because we have inverted the output, it will now deactivate when there is a fire.

Example 3: Activating an output on an input/output unit with a delayed effect

But what if we need a 10 seconds delay before activation using the same device as in the example above? The fire panel can also do this for us via the logic programming with the use of a 'timer' function.

The inputs and outputs are set up as we have seen in the previous example. The logic now will look as follows:

Figure 12: Logic Table with timer


LOGIC TABLE
1 (Input x
2)= Timer z 10
3 (Timer z
4)= Output y
5 end
User/Panel Interface Line
Alarms: Quant Faults: Quant Cond.: Quant P. ID SDZ

What does this say in English?

If the input at input table location x is true
then start timer z and time for 10 seconds
When the timer has timed out
then make the output at output location y true

So, in our case: If there is a common fire condition
then start a timer and time for 10 seconds
When the timer has timed out
then switch the output on device loop 1 device 10

Example 4: Controlling the activating of an output from a separate input unit

Scenario: When a fire alarm activates in a specific area (Area 5), we need to switch a relay that will control the sounders in that area. However, we also want to be able to disable this sounder from another location when testing is done in this area.

Equipment: Let's say we use an output on the SD2000 PCB inside the panel to switch the sounders in this area. We have a separate input from the PC to either disable this output for testing purposes, or stop the sounders once they have activated.

Programming: We need to know when we want the output to be activate, i.e. what event will we use (cause) that will cause this output to switch (effect). Then we need to know where our other input will come from…

Inputs: A fire in Area 5 will trigger the relay (Refer to section 5.1.3 AREA), but we will be able to override this trigger from a network input from the PC. (Refer to section 5.1.9 NETWORK)

On the PC:
Use a controllable Input icon. Make the input switchable, and link the status of this icon to the input on your panel as usual. The icon status will now be updated according to whether the input is true or false.

On the fire panel:

Figure 13: Area Input Menu

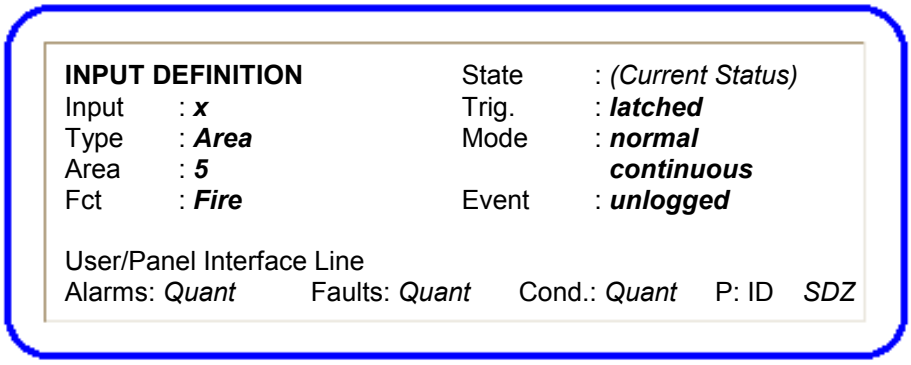
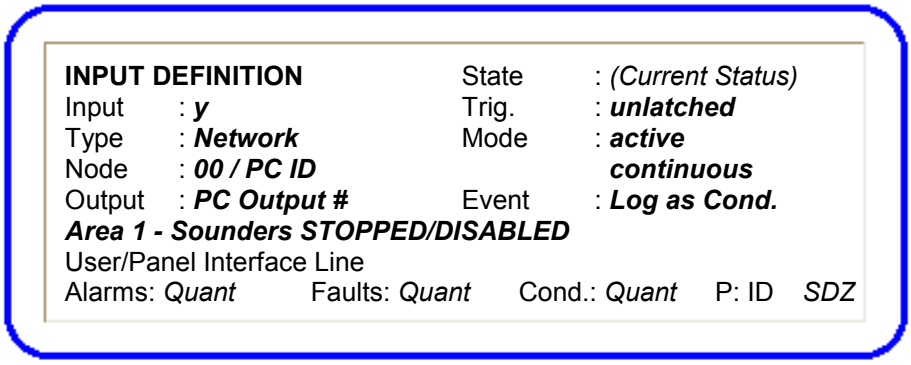


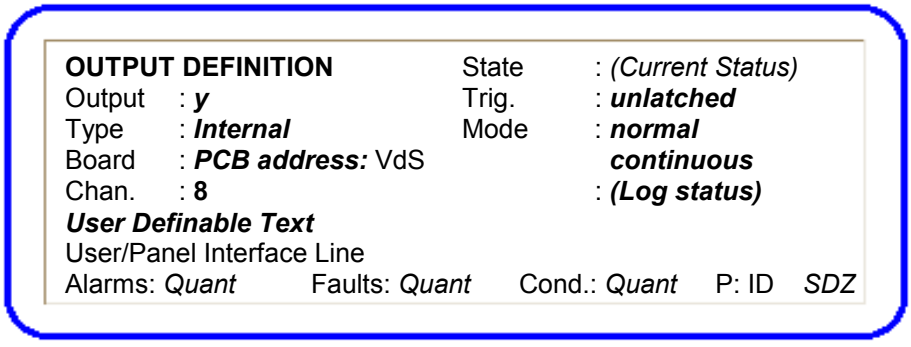
Figure 14: Network Input Menu



☞ *If we want, we may even add text to this input and **log** it **as** a **condition**. This way it will also appear as an event on the panel and in the event buffer, and you cannot forget to remove the condition once it is activated.*

Outputs: An internal output, output 8 on the SD2000 (Refer to section 6.2.4 INTERNAL)

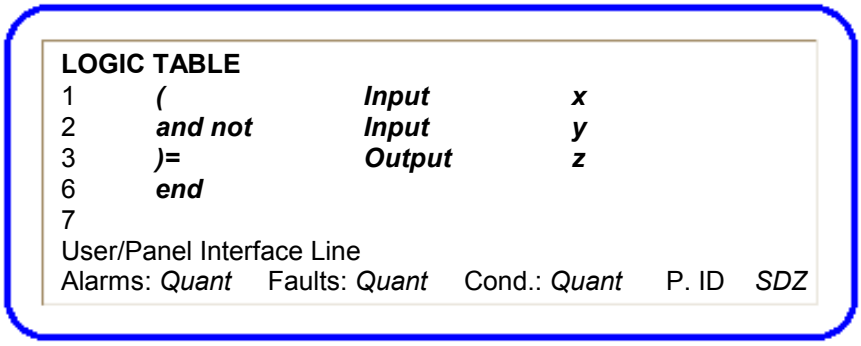
Figure 15: Internal Output



NOTE: If we want, we may even add user text to this output to describe what its function is. This way it will always know what this relay is in fact used for.

Logic: Now we have to link our ‘causes’ to our ‘effect’.

Figure 16: Logic Table



This will have the following functionality:

Condition 1: When there is a **fire in Area 5**
When the PC output is in the normal state (not disabled) and there is a fire alarm in area 5, Output z will be activated. If however the PC output is set (switched on) whilst the sounders are on, the sounder will be switched off.

Condition 1: When there is **no fire**
Switching the output on when there is no fire alarm will have the effect of disabling the output. When Input y is true, Input x will have no effect on Output z

The icon will change on the PC to another colour when it is ON, because it is linked to the true/false state of the network input. You will therefore always know when the sounders are OFF or DISABLED. There will however be no distinction between off and disabled. If you need to know this, then we can do it in another way, but you would need to use two icons and two network inputs.

Example 5: Switching two outputs from a single input

Scenario: When a fire alarm activates in a specific zone (Zone 2), we need to switch a relay that will notify the building management system to shut down air conditioning in that zone. However, we also want to switch a relay that unlatches the fire door magnets and close them.

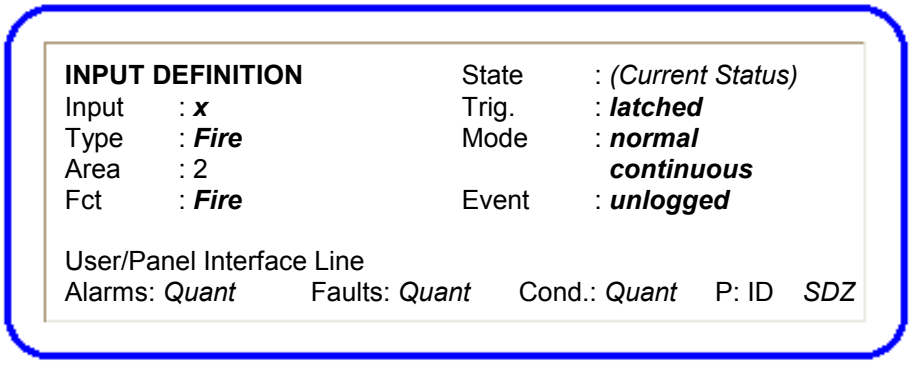
Equipment: Let's say we use two outputs on the IO2032 installed on Loop 2 Device 12.

Programming: We need to know when we want the output to be activate, i.e. what event will we use (cause) that will cause these outputs to switch (effect), and how we would like the outputs to be normally. We will need a 'fail-to safe' relay for the fire door magnets, so that the magnets will also release in the case of a critical failure, but a normal relay for the building management system, but only a single pulse. Then we need to know where our other input will come from…

Inputs: A fire in Zone 2 will trigger the relays (Refer to section 5.1.2 ZONE)

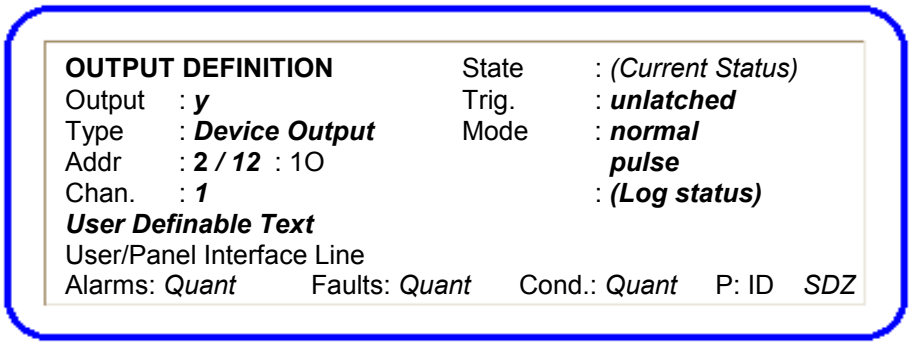
On the fire panel:

Figure 17: Fault Input Menu



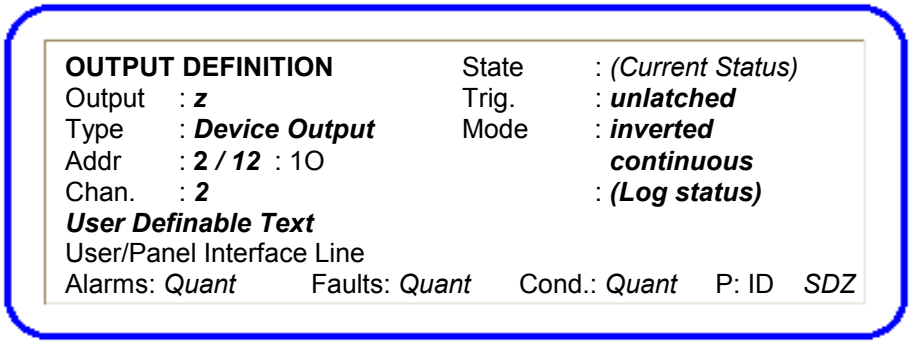
Outputs: Two device outputs, output 1 and output 2 on the IO2032 (Refer to section 6.2.5 DEVICE OUTPUT)

Figure 18: Device Output



NOTE: If we want, we may even add user text to this output to describe what its function is. This way it will always know what this relay is in fact used for.

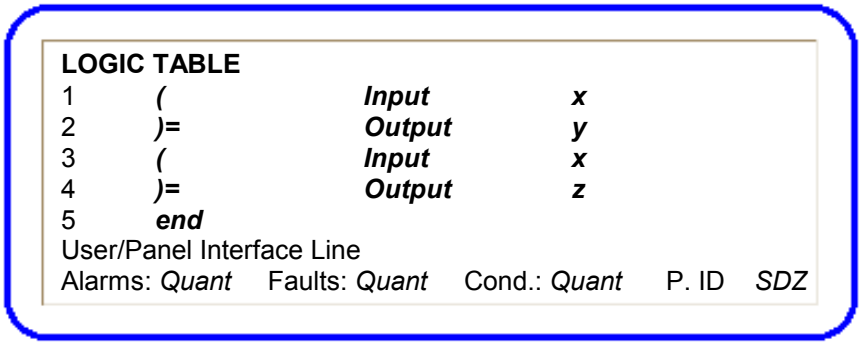
Figure 19: Device Output



NOTE: If we want, we may even add user text to this output to describe what its function is. This way it will always know what this relay is in fact used for.

Logic: Now we have to link our ‘cause’ to our ‘effects’.

Figure 20: Logic Table


LOGIC TABLE
1 (Input x
2)= Output y
3 (Input x
4)= Output z
5 end
User/Panel Interface Line
Alarms: Quant Faults: Quant Cond.: Quant P. ID SDZ

What does this say in English?

If the input at input table location x is true
then make the output at output location y true
If the input at input table location x is true
then make the output at output location y+1 true

So, in our case: If there is a fire in Zone 5
then switch the first relay on the IO2032 on address 2/12
If there is a fire in Zone 5
then switch the second relay on the IO2032 on address 2/12

☞ *The fact that Relay 1 is set to 'pulse' and relay 2 is set to 'inverted' in the Output Table, means that we don't have to worry about this in the logic.*

8. ADVANCED PROGRAMMING

Most advanced programming will be done using PCC2000, the Aritech Configuration software for programming Aritech fire panels. The following examples are extracts from site programming of various sites, and are based on the printed files from PCC2000.

Example 6: Monitoring and controlling an FD2000 (stand alone beam detector) using 2 inputs and 1 output

Scenario: Fire and fault conditions have to be reported to the panel from the FD2000. The fire and fault states should be monitored, and after a fire and/or fault is reported from the FD2000, the FD2000 should be reset from the fire panel when the fire panel is reset (or restarted).

Equipment: This is done by using a 2I/1O unit connected to the panel. For this example the device is connected on loop 2, device 126, and installed at (or inside) the FD2000 enclosure.

This is what the programming looked like …

Device Inputs:

Input	Type	Loop	Device	Channel	Trigger	Shape	Mode	Event	Input Text
11	Dev.Input	2	5	1	Unlatched	Continuous	Active	Unlogged	Fire signal from FD2000
12	Dev.Input	2	5	2	Unlatched	Continuous	Active	Unlogged	Fault signal from SD2000

Action Inputs:

Input	Type	Function	Trigger	Shape	Mode	Event
4	Action	Silence Buzzer	Unlatched	Continuous	Passive	Unlogged
13	Action	Reset	Latched	Continuous	Active	Unlogged
14	Action	Restart	Latched	Continuous	Active	Unlogged

Zone Outputs:

Output	Type	Zone	Function	Trigger	Mode	Shape	Event
11	Zone	12	Fire AUTO	Latched	Normal	Continuous	Unlogged
12	Zone	12	Fault	Latched	Normal	Continuous	Unlogged

Device Outputs:

Output	Type	Loop/Device	Channel	Linked to:	Mode	Number	Trigger	Mode	Shape	Event	Output Text
13	Dev.Out	2/5	1	Logic	None	0	Latched	Normal	Pulse	Unlogged	Reset pulse for FD2000

Logic:

Line	Operator	Operand	Number	Time	
9	(	input	11		--
10	and	timer	14		|
11	and	output	13		|
12	) =	output	11		→ Check if all these states are true, then raise the alarm
13	(	input	12		--
14	and	timer	14		|
15	and	output	13		|
16	) =	output	12		→ Check if all these states are true, then raise the fault
17	(	input	13		--
18	or	input	14		> Remember when the panel is reset or restarted
19	) =	marker	13		--
20	(	marker	13		
21	) =	timer	14	45	> Start a 45 second timer after reset or restart
22	(	marker	13		
23	) =	output	13		> Reset the FD2000 after reset or restart
25	end		0		

The tricky part of this programming is to 'mask' the fire and fault conditions sufficiently after a panel restart or reset to allow the panel to ensure that the reset/restart action of the FD2000 has in fact been executed. This is why the 45-second timer is necessary.

Example 7: Network input/output programming between panel nodes

This is part of actual programming with 1 global repeater and 5 fire panels - the fire brigade interface is connected to panel 1. The physical input, output and logic numbers are of course not important.

Scenario: Fire, fault and condition has to be reported to the fire brigade via a 3 channel I/O unit connected to panel 1 on loop 2, device 126. Every fire panel has to report a fire, fault and condition alarm to panel 1 via the network, so that this can be reported from there to the fire brigade.

This is what the programming on panel 1 looks like:

General Inputs:

Input	Type	Function	Trigger	Shape	Mode	Event
1	General	Common Fire	Latched	Continuous	Active	Unlogged
3	General	Common Cond.	Unlatched	Continuous	Active	Unlogged
20	General	Common Fault	Latched	Continuous	Active	Unlogged

These inputs capture the different alarms of panel 1 so that they can be linked to the I/O unit.

Network Inputs:

Input	Type	Output	Panel	Repeater	Trigger	Shape	Mode	Event	Input Text
4	Network	1	2	0	Latched	Continuous	Active	Unlogged	Panel 2 Fire
6	Network	3	2	0	Unlatched	Continuous	Active	Unlogged	Panel 2 Condition
7	Network	1	3	0	Latched	Continuous	Active	Unlogged	Panel 3 Fire
9	Network	3	3	0	Unlatched	Continuous	Active	Unlogged	Panel 3 Condition
10	Network	1	4	0	Latched	Continuous	Active	Unlogged	Panel 4 Fire
12	Network	3	4	0	Unlatched	Continuous	Active	Unlogged	Panel 4 Condition
13	Network	1	5	0	Latched	Continuous	Active	Unlogged	Panel 5 Fire
15	Network	3	5	0	Unlatched	Continuous	Active	Unlogged	Panel 5 Condition
60	Network	2	2	0	Latched	Continuous	Active	Unlogged	Panel 2 Fault
61	Network	5	3	0	Latched	Continuous	Active	Unlogged	Panel 3 Fault
62	Network	5	4	0	Latched	Continuous	Active	Unlogged	Panel 4 Fault
63	Network	5	5	0	Latched	Continuous	Active	Unlogged	Panel 5 Fault

These inputs are received on this panel from every other panel in the network so that they can be linked to the I/O unit. Note that the Output number on each transmitting panel is also specified.

Device Outputs:

Output	Type	Loop/Device	Channel	Linked to:	Trigger	Mode	Shape	Event	Output Text
1	Dev.Out	2/126	1	Logic	Latched	Normal	Continuous	Unlogged	Report Fire
2	Dev.Out	2/126	3	Logic	Unlatched	Normal	Continuous	Unlogged	Report Condition
3	Dev.Out	2/126	2	Logic	Unlatched	Normal	Continuous	Unlogged	Report Fault

These are the physical outputs on the 3-channel I/O unit.

Logic:

Line	Operator	Operand	Number	Time	
1	(	input	1		--
2	or	input	4		|
3	or	input	7		> All the Fire alarms in the system 'or'ed together
4	or	input	10		|
5	or	input	13		--
6	) =	output	1		
7	(	input	3		--
8	or	input	6		|
9	or	input	9		> All the Conditions in the system 'or'ed together
10	or	input	12		|
11	or	input	15		--
12	) =	timer	4	180	Output delayed for 3 minutes
13	(	timer	4		
14	) =	output	2		
15	(	input	20		--
16	or	input	60		|
17	or	input	61		> All the Faults in the system 'or'ed together
18	or	input	62		|

Line	Operator	Operand	Number	Time	
19	or	input	63		--
20	) =	output	3		
21	end				

And all that remains is to simply link all the inputs to their correct outputs on the 3-channel I/O unit.

This is what panel 2 (and all other panels') programming looks like:

General Inputs:

Input	Type	Function	Trigger	Shape	Mode	Event
1	General	Common Fire	Latched	Continuous	Active	Unlogged
3	General	Common Cond.	Unlatched	Continuous	Active	Unlogged
20	General	Common Fault	Latched	Continuous	Active	Unlogged

These inputs capture the different alarms of this so that they can be sent to panel 1.

Network Outputs:

Output	Type	Input	Panel	Repeater	Trigger	Mode	Shape	Event	Output Text
1	Network	4	1	0	Latched	Normal	Continuous	Unlogged	Report Fire to Panel 1
2	Network	60	1	0	Latched	Normal	Continuous	Unlogged	Report Fault to Panel 1
3	Network	6	1	0	Unlatched	Normal	Continuous	Unlogged	Report Condition to Panel 1

These outputs pass the messages to panel 1. Note how the input numbers where they must go on panel 1 is listed with each output.

Logic:

Line	Operator	Operand	Number	Time
1	(	Input	1	
2	) =	Output	1	
3	(	Input	2	
4	) =	Output	2	
5	(	Input	3	
6	) =	Output	3	
7	end			

And all that remains is to simply link all the inputs to their correct outputs on the network.

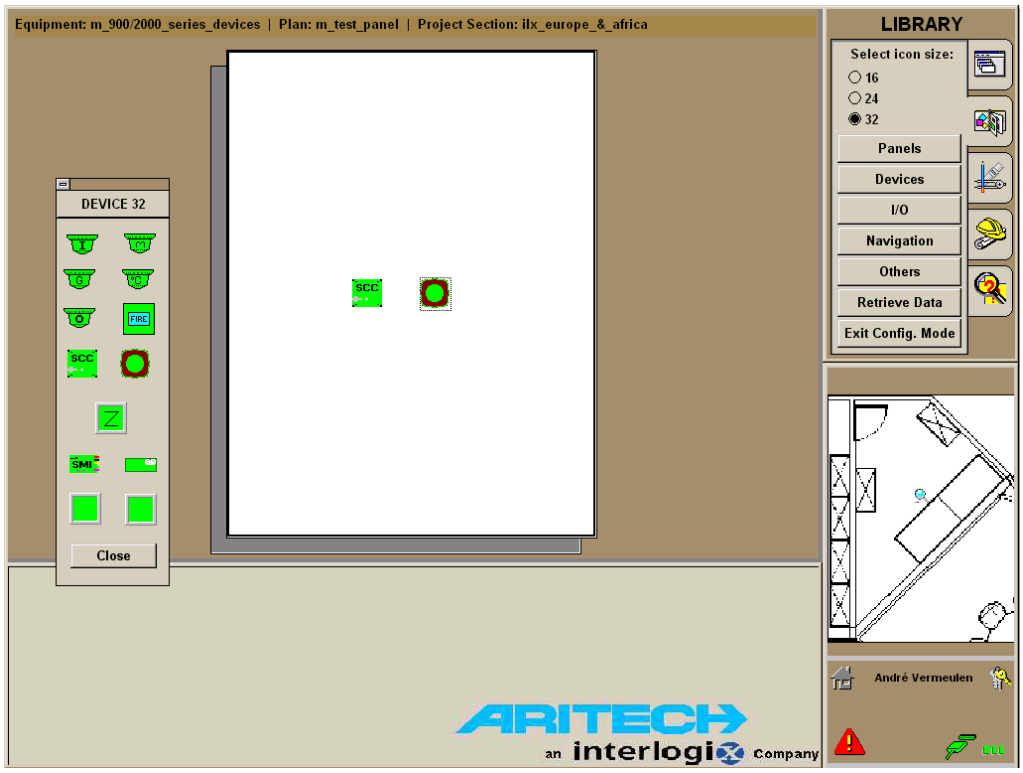
Example 8: Network programming between PC node and a panel node

Scenario: The operator needs to control a sounder from the PCG2001 software.

Equipment: For this we will use a SCC (Sounder Circuit Controller) icon from the devices toolbar: (We can also use the Sounder Icon from the devices toolbar - right next to the SCC icon)

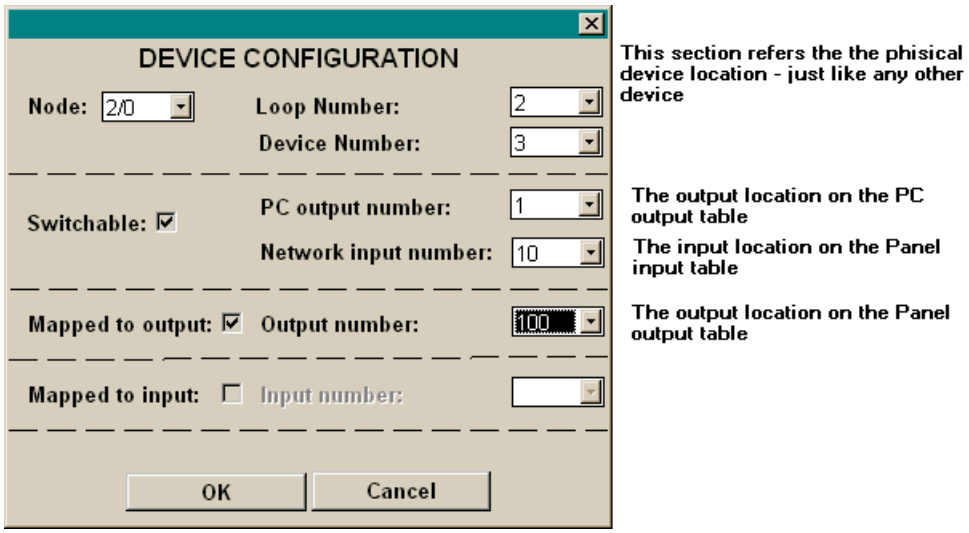
1. Place the sounder device on the plan as required.

Figure 21: Device Control Icons



2. Configure the icon on the PC as required.

Figure 22: Control Device configuration

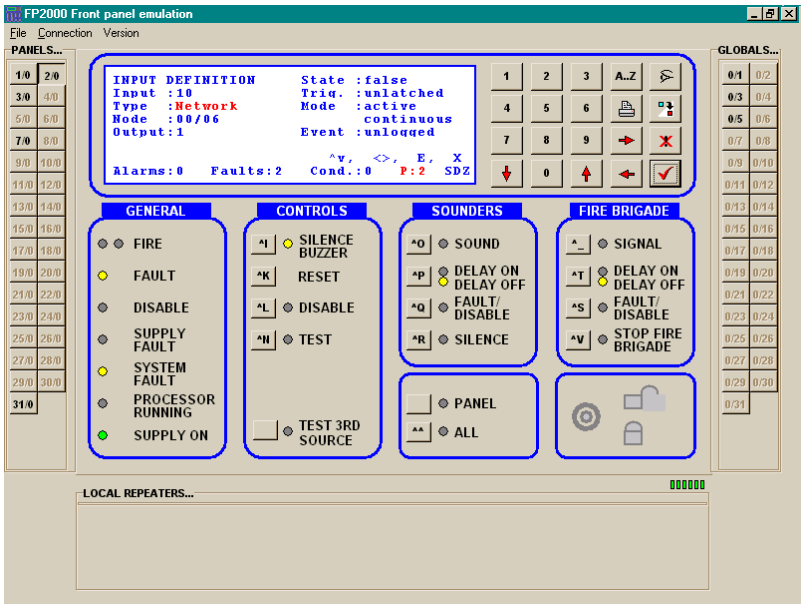

DEVICE CONFIGURATION
Node: 2/0
Loop Number: 2
Device Number: 3
Switchable:
PC output number: 1
Network input number: 10
Mapped to output:
Output number: 100
Mapped to input:
Input number:
OK
Cancel
This section refers the the phisical device location - just like any other device
The output location on the PC output table
The input location on the Panel input table
The output location on the Panel output table

This is all that is required on the PC.

The panel must be programmed as follows:

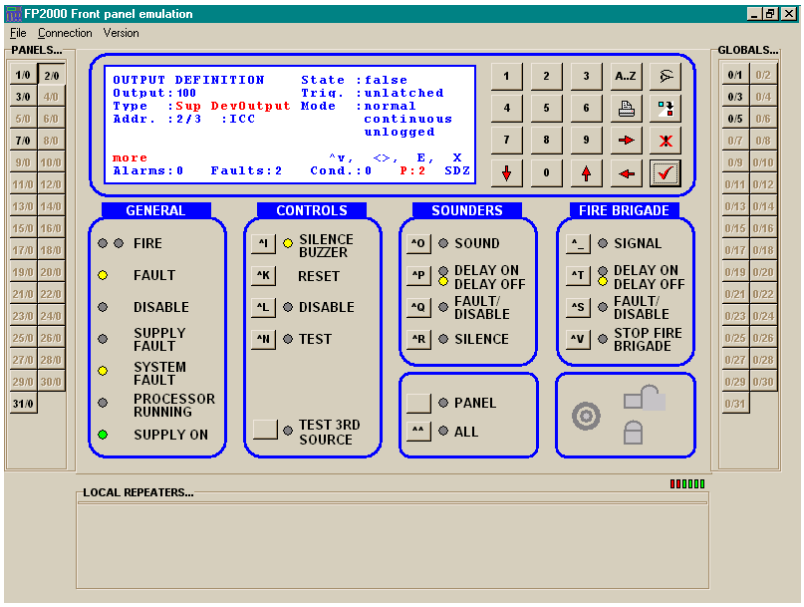
3. Go to the input table of the panel. Go to the input number referred to in the panel input in 2 above. Configure this input as a network input. The Node ID here will of course refer to the PC ID, and the output number will refer to the PC output number programmed in 2 above. Ensure that this input is 'unlatched', else it cannot be switched off without a reset from the fire panel.

Figure 23: Panel Input configuration



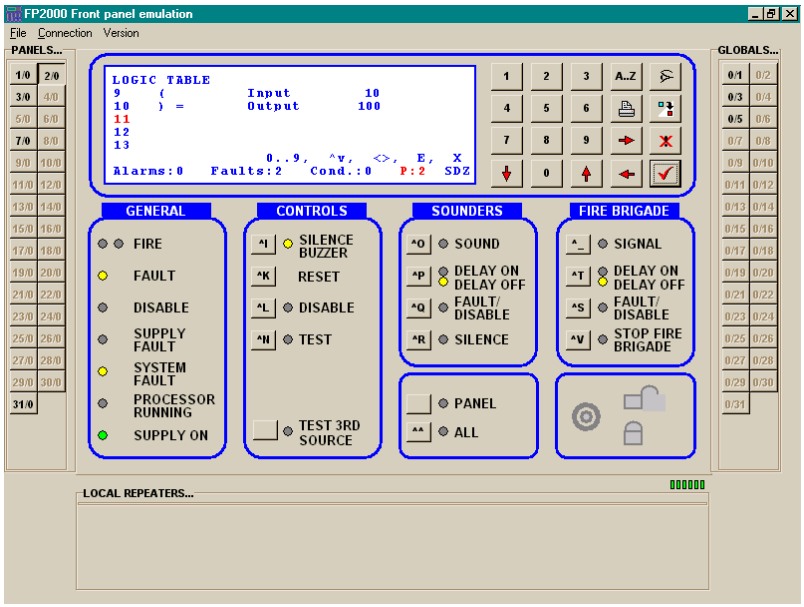
4. Go to the output table of the panel. Go to the output number programmed in 2 above as the Panel output number. Configure this as a supervised device output - since we are working with a sounder controller here. The device location here will of course refer to the connected position of this sounder, and will match the address referred to in (2) above. Ensure that this output is 'unlatched', else it cannot be switched off without a reset from the fire panel.

Figure 24: Panel Output configuration



5. Go to the logic table, and link the input in 3 and the output in 4 together.

Figure 25: Panel Logic configuration



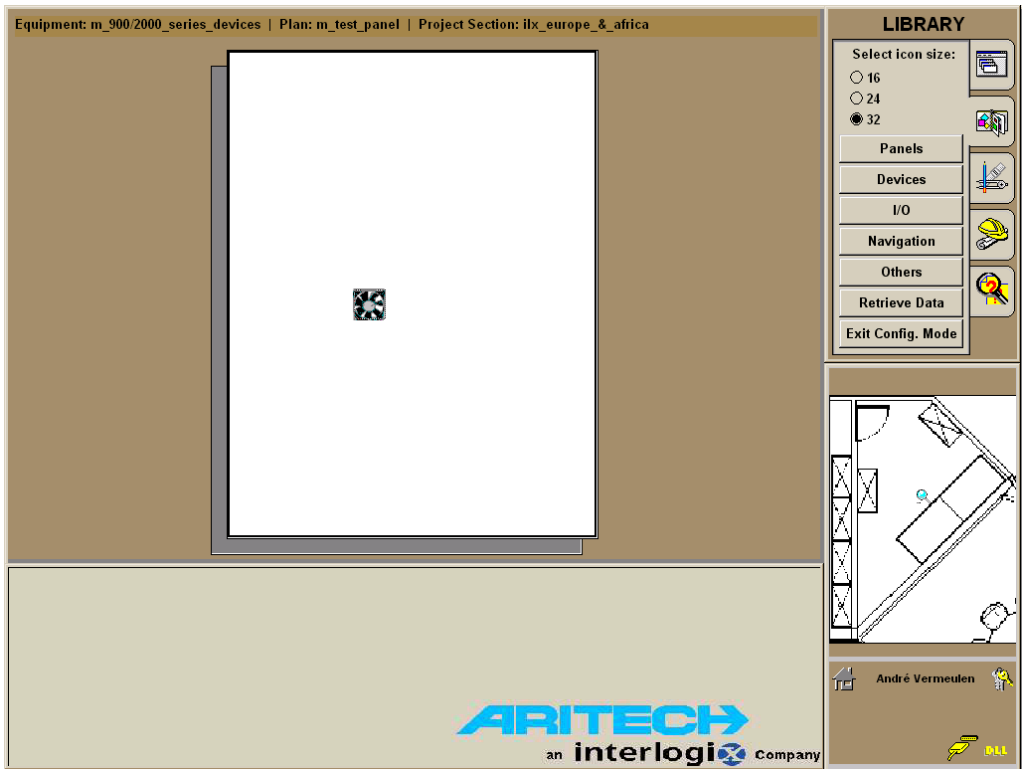
Exit all menus, ensure that you have no faults on the panel, and you're ready to go! This icon will now allow the status from the ICC (panel 2 loop 2 device 3 in this case) to be displayed, but will also monitor the 'true/false' state of output 100 on panel two, showing whether this output is in fact active or not. Simply right-click on the device to turn the output 'on' or 'off'. The status is updated on the screen about once every 10 seconds.

For the second scenario we will not use an icon that will show the status of the device, but simply whether the output is on or off when controlled from the PC. Here we will use the fan icon to show and switch an output on the SD2000 in the panel.

On the PC

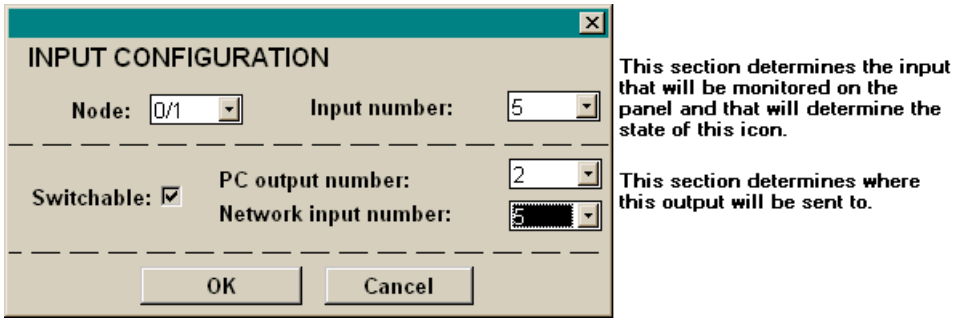
1. Place the icon from the 'Input' configuration of the I/O Toolbar. Any icon from this input toolbar will do the same thing.

Figure 26: Control Monitor icons



2. Configure the icon on the PC as required. Normally one will monitor the same input that we are switching, but this is at your own discretion.

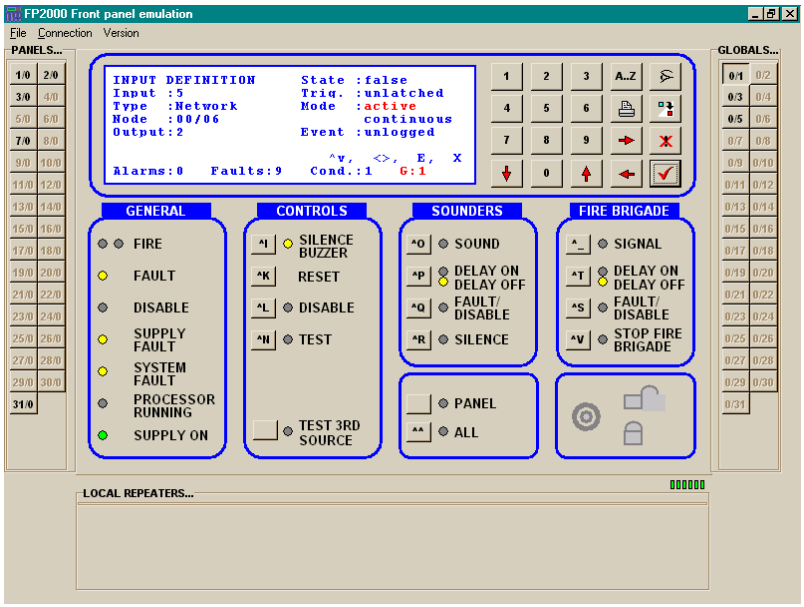
Figure 27: Control Monitor configuration


INPUT CONFIGURATION
Node: 0/1
Input number: 5
Switchable:
PC output number: 2
Network input number:
OK
Cancel
This section determines the input that will be monitored on the panel and that will determine the state of this icon.
This section determines where this output will be sent to.

On the panel:

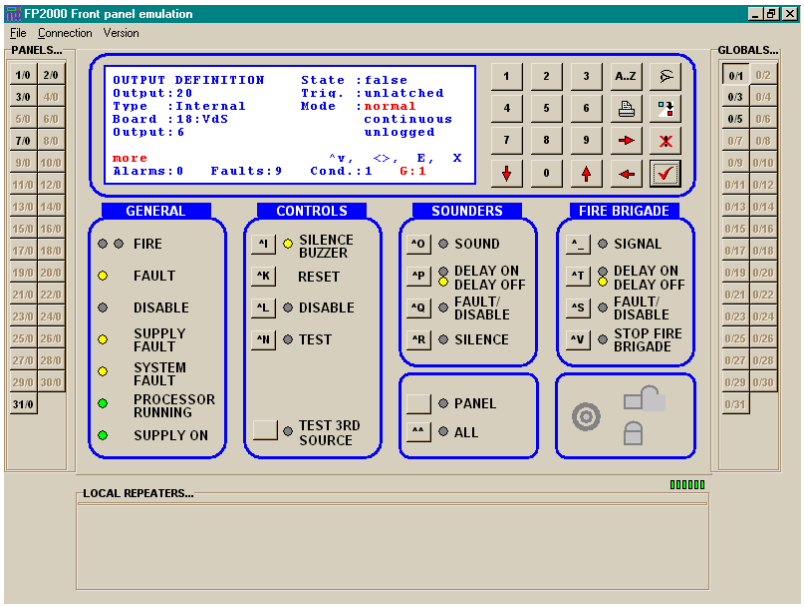
3. Go to the input table of the panel. Go to the input number referred to in the panel input in 2.2 above. Configure this input as a network input. The Node ID here will of course refer to the PC ID, and the output number will refer to the PC output number programmed in 2.2 above. Ensure that this input is 'unlatched', else it cannot be switched off without a reset from the fire panel.

Figure 28: Panel Input configuration



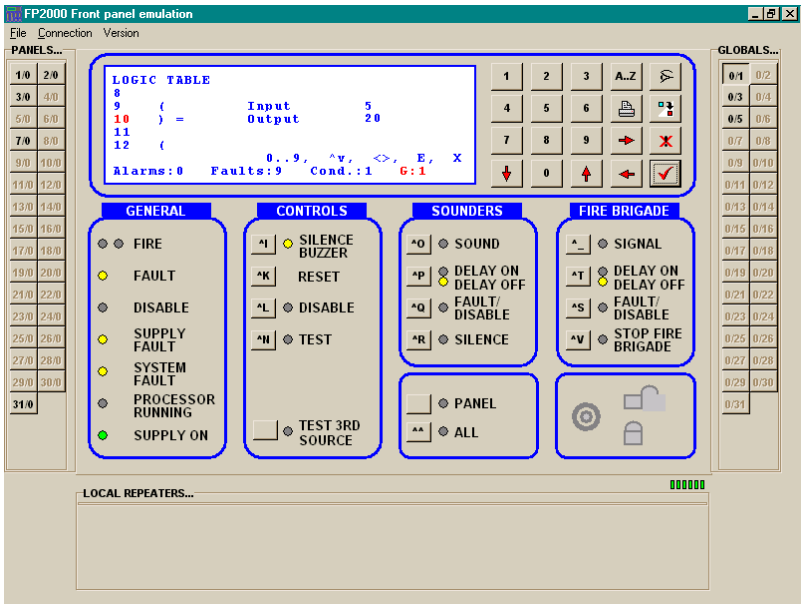
4. Go to the output table of the panel. Go to any output number and configure this as an internal output - since we are working with a SD2000 controller here. Ensure that this output is 'unlatched', else it cannot be switched off without a reset from the fire panel.

Figure 29: Panel Output configuration



5. Go to the logic table, and link the input in 2.3 and the output in 2.4 together.

Figure 30: Panel Logic configuration



Exit all menus, ensure that you have no faults on the panel, and you're ready to go! This icon will now allow the status of the output on the SD2000 PCB, showing whether it is in fact true or not. Simply right-click on the device to turn the output 'on' or 'off'.

That's it! You'll be able to control input 5 now on this panel, and potentially it can switch anything, depending on how you do your logic. In this case it is switching relay 6 on the SD2000 PCB. The status is updated on the screen about once every 10 seconds.

Example 9: Gas Control Unit Programming

Scenario: In this example we will show that it is possible to turn the panel into a GAS control panel. We do however emphasise that this example is used for explanation purposes only and is NOT normally recommended as a true solution in the field, since the outputs here are all unsupervised and therefore not suitable for this type of application.

Equipment: Here we will use as inputs, the internal inputs inside the panel, both on the FEP PCB as well as on the SD2000 PCB. The panel has to be in an operation mode other than VdS. As the main controller, we will use an FM808, the 8-output current loop device.

First we'll do the monitoring of the gas status i.e. the user interface.

Internal Inputs:

Input	Type	Brd.Addrs	Channel	Adr 2	Trigger	Shape	Mode	Event	Input text
1	Internal	24	1		Unlatched	Continuous	Active	As Cond.	Zone 100 Gas control in MANUAL MODE
2	Internal	17	5		Latched	Continuous	Short	As Fault	Emergency stop activated
3	Internal	17	6		Unlatched	Continuous	Short	Unlogged	Discharge confirmed
4	Internal	17	7		Unlatched	Continuous	Short	As Fault	Container pressure low
5	Internal	17	8		Latched	Continuous	Short	As fire	Zone 100 - Manual extinguish command

Here is the zone that we will protect…

Zone Inputs:

Input	Type	Zone	Function	Trigger	Shape	Mode	Event
6	Zone	100	Fire	Latched	Continuous	Active	Unlogged

Input	Type	Zone	Function	Trigger	Shape	Mode	Event
7	Zone	100	Co-incidence	Latched	Continuous	Active	Unlogged

Then we'll do the outputs for the field controller…

Current Loop Outputs:

Output	Type	Address	Output	Linked to:	Mode	Number	Trigger	Mode	Shape	Event	Output text
1	CL Device	1	1	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	Gas discharge output
2	CL Device	1	5	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	Indicator gas discharged
3	CL Device	1	2	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	First alarm bells
4	CL Device	1	6	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	First alarm indicator
5	CL Device	1	3	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	Second alarm siren
6	CL Device	1	7	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	Second alarm indicator
7	CL Device	1	4	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	Zone 100 ALARM
8	CL Device	1	8	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	Start extinguishing

and for the user interface.

Internal Outputs:

Output	Type	Brd.Addrs	Channel	Linked to:	Mode	Number	Trigger	Mode	Shape	Event	Output text
9	Internal	17	5	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	First alarm output
10	Internal	17	6	Logic	None	0	Unlatched	Normal	Continuous	Unlogged	Second alarm output

Logic:

Line	Operator	Operand	Number	Time	
1	(				
2	(	input	7		If there is co-incidence in zone 100
3	and	not input	1		and the control is NOT in manual mode,
4	)				
5	or	input	5		or there is a manual extinguishing activation command,
6	) =	marker	1		set this marker.
7	(	marker	1		When this marker is set,
8	) =	timer	1	15	start a timer for 15 seconds.
9	(	timer	1		When timer one has timed out
10	and	not input	2		and the emergency stop has not been activated,
11	) =	output	1		discharge this gas.
12	(	input	6		If there is a fire in zone 100
13	and	not input	1		and the control is NOT in manual mode,
14	) =	marker	6		set this marker.
15	(	input	3		When the discharge has been confirmed,
16	) =	output	2		light the discharge indicator lamp.
17	(	marker	6		When this marker is set,
18	) =	output	3		ring the warning bells for first fire.
19	(	marker	6		When this marker is set,
20	) =	output	4		light the first alarm indicator.
21	(	marker	1		When this marker is set,
22	) =	output	5		ring the alarm bells for second fire
23	(	marker	1		When this marker is set,
24	) =	output	6		light the second alarm indicator.
25	(	marker	1		When this marker is set,
26	) =	output	7		light the alarm indicator for this zone.
27	(	marker	1		When this marker is set,
28	) =	output	8		light the indicator for discharge timer started.
29	(	marker	6		When this marker is set,
30	) =	output	9		switch the first fire alarm output.
31	(	marker	1		When this marker is set,
32	) =	output	10		switch the second fire alarm output.
33	end				